1 DATA DESCRIPTION AND RESEARCH QUESTION

Distributed data analysis is a practice that stores data where it lives empowering business analysis through a single point of access. The aim of this project is to generate value and insight from the processing of the data by implementing several analytic methods and algorithm, evaluating them and comparing the effectiveness of the adopted approaches.

1.1 DATA

The dataset for this project was sourced from The Murder Accountability Project. It is the most complete database of homicides in the United States currently available. This dataset includes murders from the FBI's Supplementary Homicide Report from 1976 to the present and Freedom of Information Act data on more than 22,000 homicides that were not reported to the Justice Department. The Murder Accountability Project makes available to police and the general public all of the data files it uses for this Website .(Murder Accountability Project: Data & Docs (murderdata.org).The data set consist of the following description

ID – Unique record identifier generated

CNTYFIPS – The Census Bureau's Federal Information Processing Standards (FIPS) code designating the state and county of the reporting law enforcement agency.

ORI – The alphanumeric variable describing the Originating Agency making the report.

STATE – The alphanumeric variable describing the state of the Originating Agency making the report.

AGENCY – The alphanumeric variable describing the name of the law enforcement agency making the report

AGENTYPE – The one-digit numeric code describing the type of law enforcement agency making the report.

SOLVED – MAP-generated indicator whether Offender was identified at time report was made (SOLVED=1) or not identified (SOLVED=0). Numeric single digit format.

YEAR – Year of homicide

MONTH – The month of homicide occurrence or when the victim's body was recovered.

INCIDENT – A three-digit number describing the case number within the month in which a homicide occurred.

ACTIONTYPE – A numeric variable describing the nature of the report received.

HOMICIDE – An alphanumeric variable defining whether the report was "A" = "Murder or Non negligent manslaughter" or "B" = "Manslaughter by Negligence."

SITUATION – An alphanumeric variable defining whether the crime had a single victim or multiple victims and whether there was a single offender, multiple offenders or the number of offenders was unknown.

VICAGE – A three-digit numeric variable describing the age in years of the victim.VICSEX – An alphanumeric variable representing whether the victim was "M" =" Male" or "F" =" Female" or "U" indicating "Unknown" gender, usually for conditions in which incomplete remains were recovered.

VICRACE – An alphanumeric variable representing victim race.

VICETHNIC – An alphanumeric variable representing victim ethnicity.

OFFAGE – A three-digit numeric variable describing the age of the offender.

OFFSEX – An alphanumeric variable representing whether the offender was “M” =” Male” or “F” =” Female” or “U” indicating “Unknown” gender

OFFRACE – An alphanumeric variable representing offender’s race.

OFFETHNIC – An alphanumeric variable representing offender’s ethnicity.

WEAPON – A two-digit numeric variable representing the weapon used in the crime.

RELATIONSHIP – An alphanumeric variable describing the relationship between the victim and the offender, if any

CIRCUMSTANCES – A two-digit numeric variable representing the circumstances (or theory) for the crime.

SUBCIRCUM – A single-digit alphanumeric variable describing several conditions in which the victim is reported to have been a criminal offender.

VICCOUNT – The number of additional victims (not counting the victim included in the current record) included in the Supplementary Homicide Report’s incident record.

OFFCOUNT – The number of additional offenders (not counting the offender included in the current record) included in the Supplementary Homicide Report’s incident record.

FSTATE – A two-digit alphanumeric variable representing the state in which a homicide was reported.

MSA – An eight-digit numeric variable representing the Census Bureau’s Federal Information Processing Standards (FIPS) code for the Metropolitan Statistical Area from which a record was reported.

1.2 RESEARCH QUESTION

The research question: What are the factors associated with predicting offender demographics, specifically based on age and gender, in the context of homicide cases in the United States, utilizing victim demographics, case characteristics, and organizational factors?

The aim of the research question is to predict the offender’s demographic that is age and gender by considering the factors like victim’s demographic , the characteristics of the case and organizational factors.

2.DATA PREPARATION AND CLEANING

2.1 DATA PREPARATION

The objective of data preparation and cleaning process is to ensure that dataset is accurate and ready for analysis. The data preparation and cleaning is conducted in R .

- The first step was to download the necessary R packages and load the data set ‘homicide’ which is a csv file as a data frame in R.
- The dataset named homicide was sub stetted from year greater than 2000 as the subsample where crime records from 2000-2022 were considered for this report and renamed it as crime cleaned.
- The data frame was explored using the view and head functions and dimensions were obtained through nrow and ncol functions. The data set has 30 variable and 390689 observations.
- The summary and structure functions were used to make sure that R has read all the variables correctly. Upon the inspection, the data set was found to contain unnecessary variables, missing values, categorical values were not read as factor levels , column names were not significant .So the data set was cleaned

2.2 DATA CLEANING

The data cleaning steps which were as follows:

- The variables such as 'ID','Ori','Agency','Incident','Source','File Date','MSA' were removed because they were not required for the data analysis for predicting the offender age and sex
- The variable 'CNTYFIPS' was modified to create a new variable to have a county name where crime happened and some of the variable names were renamed for better understanding.
- The categorical variables were converted into factors with defined levels for accurate representation and enhancing analytical usability .
- After generating frequency table for each categorical variables Victim_sex','Victim_race','Victim_ethnicity','Offender_sex',"Offender_race','Offender_ethnicity' have a 'Unknown' category, so they were recategorized as missing as their percentages were low.
- The ethnicity for both offender and victim were removed because 50% of the data for this column was missing.
- Selected categorical variables levels were renamed due to lengthy description or unnecessary information (refer code section).
- The numerical columns were explored, and the validation rules were applied to assess the variable constraints.
- Certain issues were identified from the validation, such as lot of values were found as ‘999’ in Victim age and there were outliers detected in Offender age with values such as age>99.
- To address these issues, ‘999’ was indicated as missing in Victim age and implausible records in Offender age were removed.
- The duplicate values were explored and removed.
- No missing values were found in other variables except of Victim age.
- All the missing values were removed.
- A new target or dependent variable ‘Offender demo’ which gives the offender demographics was created by grouping Offender age and Offender gender by forming 6 levels such as "<18M", "<18F", "19-55M", "19-55F", ">56M", ">56F."
- The cleaned dataset was written into a csv file as crime cleaned.

The data preparation and cleaning has resulted into a high-quality dataset that is well suited for analysis.

3. EXPLORATORY DATA ANALYSIS

The exploratory data analysis is done on the cleaned data set ‘crime cleaned’ to gain insights into the dataset characteristics, uncover patterns and identify relationships between variables.

Summary statistics: The summary statistics were performed for the variables. Victim age, Year ,Add Victim Count, and Add offender count which were the numerical variables retrieved descriptive statistics such as mean, median, standard deviation was retrieved.

Univariate Analysis: Univariate analysis was performed to explore the distribution of individual variables.

Histograms were used to analyze the distribution of numerical variables Victim age, Offender age , add victim count , and add offender count. Left skewness in both victim age and offender age.

Bar plots were used to analyses the distribution of categorical variables Agency type , Solved, Crime type, Crime Status, Victim Sex, Victim Race, Offender Race, Relationship, Weapon, Crime Cause, Victim prior offense state, State, Month, Year ,Report.

The bars indicated higher frequency of murder and non negligent manslaughter in Crime type, male in Victim sex , white in Victim Race as well as in Offender Race, the use of handgun as Weapon in crimes, most of the crimes happened in Texas in State variable, and relationship were unknown, and most of the crimes were Solved and the dependent variable

Bivariate Analysis: Bivariate analysis was performed to explore the relationship between the independent variables and the dependent variable .Grouped Bar plots were used for the categorical variables and the target variable Offender demographic(Offender demo).

The bar plot of Offender_demo and Crime_type indicated that frequency of data points were distributed between 19-55M and Murder /Non negligence manslaughter.

The bar plot of Offender_demo and Victim sex indicated that frequency of data points were distributed between 19-55M and Male.

The bar plot of Offender_demo and Weapon indicated that frequency of data points were distributed between 19-55M and Handgun

The bar plot of Offender_demo and Victim race indicated that frequency of data points were distributed between 19-55M and White and also Black.

The bar plot of Offender_demo and Weapon indicated that frequency of data points were distributed between 19-55M and Handgun.

The bar plot of Offender_demo and Relationship indicated that frequency of data points were distributed between 19-55M and Unknown.

The bar plot of Offender_demo and Agency type indicated that frequency of data points were distributed between 19-55M and Muncipal office

The bar plot of Offender_demo and Victim prior indicated that frequency of data points were distributed between 19-55M and Not specified.

From additional graphs , the bar plot of Crime type and relationship indicated frequency of points were distributed between Murder and Acquaintance.

The bar plot of Crime type and State had a distribution of points between Murder and California.

The box plot of Offender demo and Victim age indicated distribution of data between >56M and in the range of 50-60 victim age.

Multivariate analysis: Correlation coefficients were determined for the numerical variables using gcorr .

An unsupervised method Principal Component Analysis was used in this project. Principal Component Analysis is used for dimensionality reduction that is reduction of features and variables.The variance plot of principal component analysis shows the percentage of the variance explained by first 10 Pcs.The first PCA showed the most percentage variance in the plot.The proportion of variance plot depicts the similar amount in scale.The cumulative PEV plot gives the total of how much variance can the all the PCs can explain.The bi plot of Principal component analysis also have been analysed and gives much importance on the pca variance.

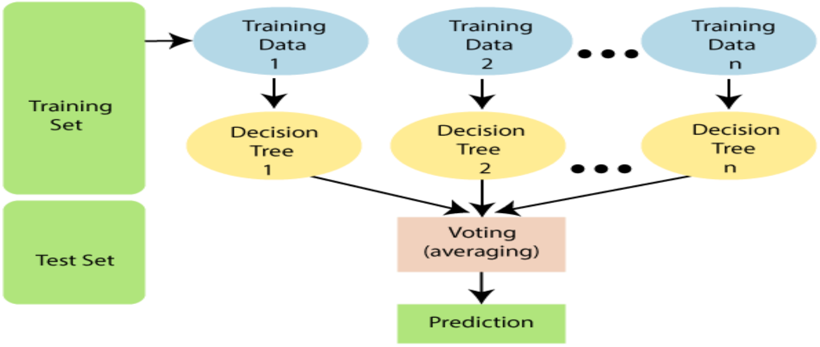
An additional method called Factor Analysis of Mixed Data was used to generalize the Principal Component analysis.The FAMD produced the same interpretations except in does not require categories to be encoded like the PCA.

4. MACHINE LEARNING PREDICTION

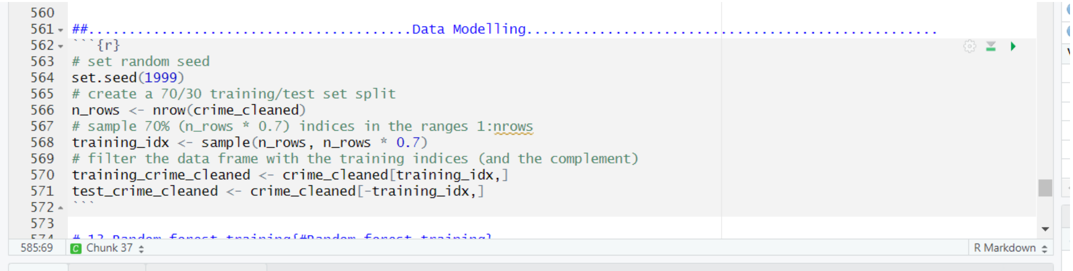
Machine learning is a field of technology that allows computer to automatically learn from previous data. It is used building mathematical models and predicitons based on historical data or information.

Random forest is one of the machine learning methods that have been used in this project .This method is used for both classification and regression . Hence this method is used to model the Crime data set for prediction of the Offender demographics that is age and gender which is a classification dependent variable.

Working of the Random Forest Method: Random forest classifier method works by splitting the data set into various sub sets and each subsets contains a number of decision trees and then the average of them is taken to predict the accuracy of the dataset. Random forest takes prediction from each tree instead of just one decision tree and based on the majority of the predictions ,it predicts the final output. More number of trees leads to more accuracy

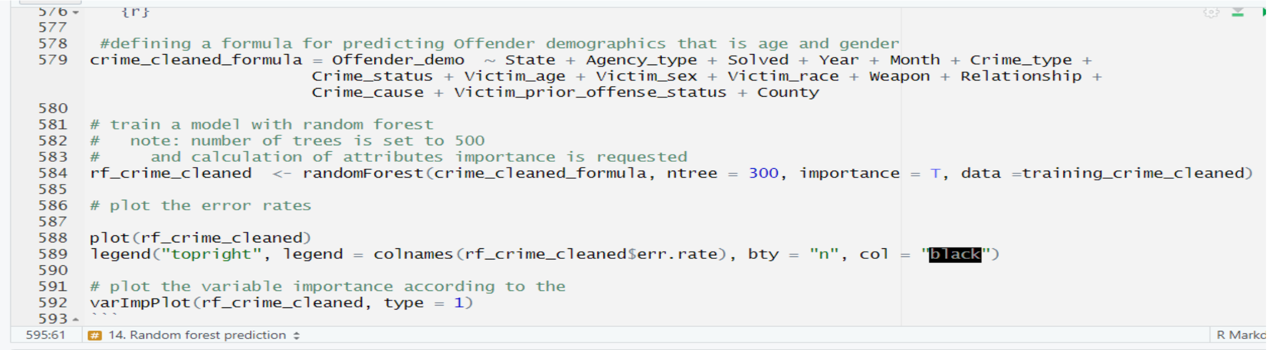


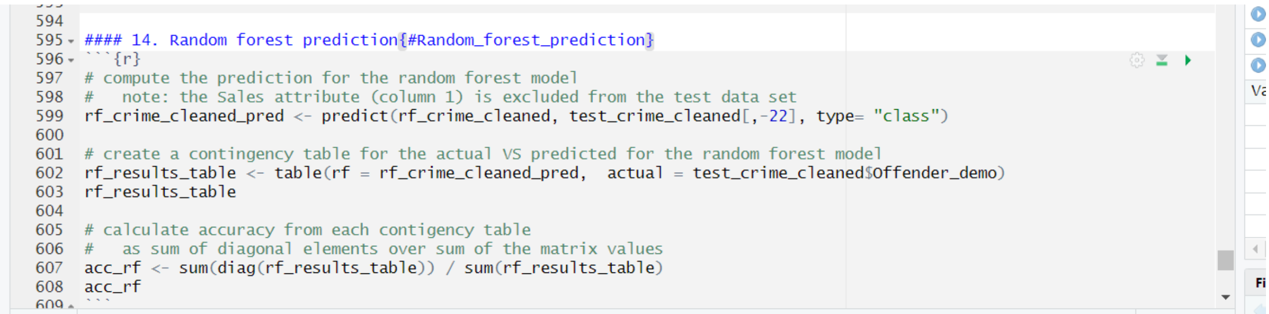
Data Modelling: Random forest classifier modelling for this dataset Crime (Crime_cleaned) is executed in R. In the first section of the modelling, in the code the data set is split into 70 % training and 30 % test subsets.

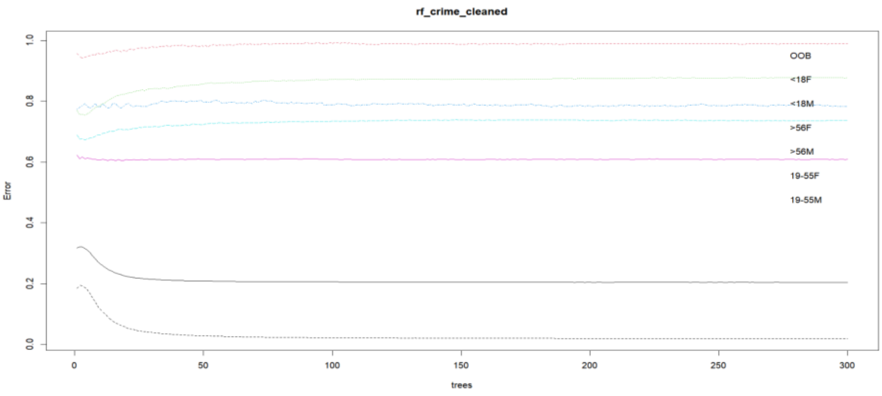


~~~
##.....................................Data Modelling..................................................
```{r}
# set random seed
set.seed(1999)
# create a 70/30 training/test set split
n_rows <- nrow(crime_cleaned)
# sample 70% (n_rows * 0.7) indices in the ranges 1:nrows
training_idx <- sample(n_rows, n_rows * 0.7)
# filter the data frame with the training indices (and the complement)
training_crime_cleaned <- crime_cleaned[training_idx,]
test_crime_cleaned <- crime_cleaned[-training_idx,]
```

# 12 Random forest training{#Random_forest_training}
~~~

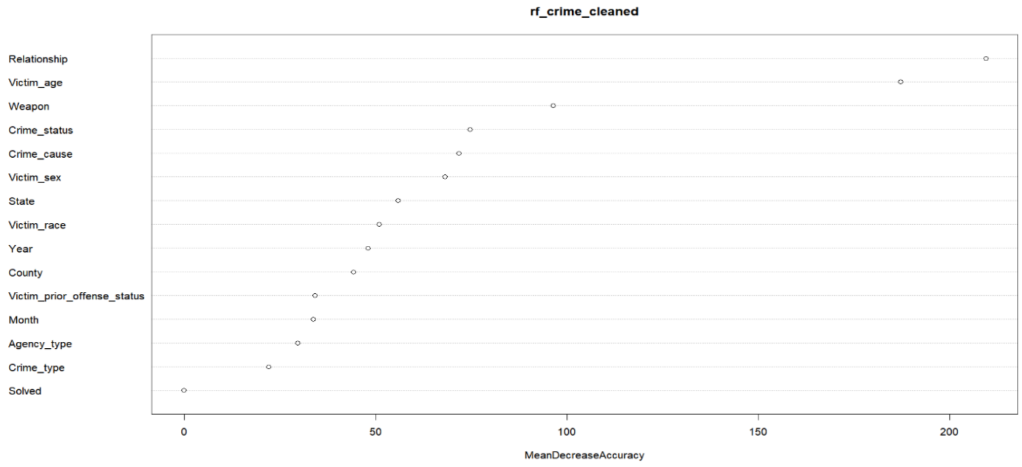
A formula is defined for predicting the dependent variable. Using the formula the model is trained using the random forest on the training data set , and with trees set to 300.The error rates and variable importance in the dataset are plotted.



~~~
```{r}

 #defining a formula for predicting Offender demographics that is age and gender
crime_cleaned_formula = Offender_demo  ~ State + Agency_type + Solved + Year + Month + Crime_type +
                             Crime_status + Victim_age + Victim_sex + Victim_race + Weapon + Relationship +
                             Crime_cause + Victim_prior_offense_status + County

# train a model with random forest
#   note: number of trees is set to 500
#      and calculation of attributes importance is requested
rf_crime_cleaned  <- randomForest(crime_cleaned_formula, ntree = 300, importance = T, data =training_crime_cleaned)

# plot the error rates

plot(rf_crime_cleaned)
legend("topright", legend = colnames(rf_crime_cleaned$err.rate), bty = "n", col = "black")

# plot the variable importance according to the
varImpPlot(rf_crime_cleaned, type = 1)
~~~
```

595:61
14. Random forest prediction
R Marko

Then random forest prediction is done on test data set , contingency table is created for actual and predicted set and accuracy results are printed.



```

#### 14. Random forest prediction{#Random_forest_prediction}
```{r}
# compute the prediction for the random forest model
#   note: the Sales attribute (column 1) is excluded from the test data set
rf_crime_cleaned_pred <- predict(rf_crime_cleaned, test_crime_cleaned[,-22], type= "class")

# create a contingency table for the actual VS predicted for the random forest model
rf_results_table <- table(rf = rf_crime_cleaned_pred,  actual = test_crime_cleaned$Offender_demo)
rf_results_table

# calculate accuracy from each contigency table
#   as sum of diagonal elements over sum of the matrix values
acc_rf <- sum(diag(rf_results_table)) / sum(rf_results_table)
acc_rf
```
```

In the error plot graph below depicts the relationship between number of trees and the performance metrics such as classification rate. The error rate determines the optimal number of trees, hence here 300 is set to be the optimal number of trees.


```

In the VarImp graph below, the variables Relationship , Victim age , and weapon indicates higher importance in the prediction for the model.

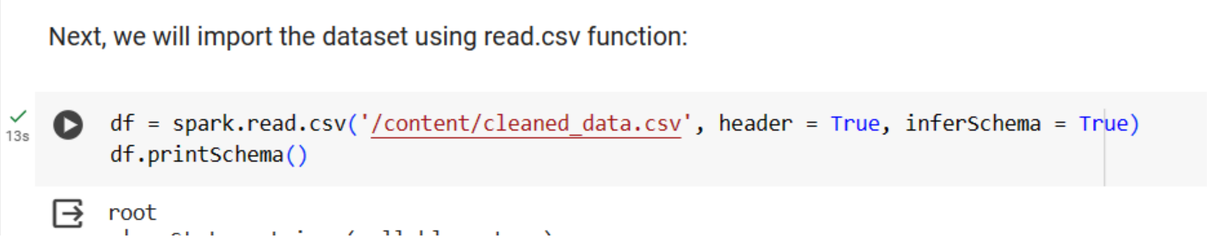


# 5 HIGH PERFORMANCE COMPUTATION METHOD

Apache Spark is the high performance computation method used in this project. Apache Spark is an open source computing framework for large scale data processing. The machine learning method Random Forest Classifier is executed in PySpark . Pyspark is the Python API for Apache Spark which enables to perform high scale data processing in a distributed environment using python and also provides PySpark shell for analyzing the data.( spark.apache.org, n.d).

Google colab is used to execute the python code for Random Forest classifier method .In google colab , all Pyspark libraries are installed and Pyspark session is created.

Then the data set which is in the form of csv file is loaded into a data frame df.



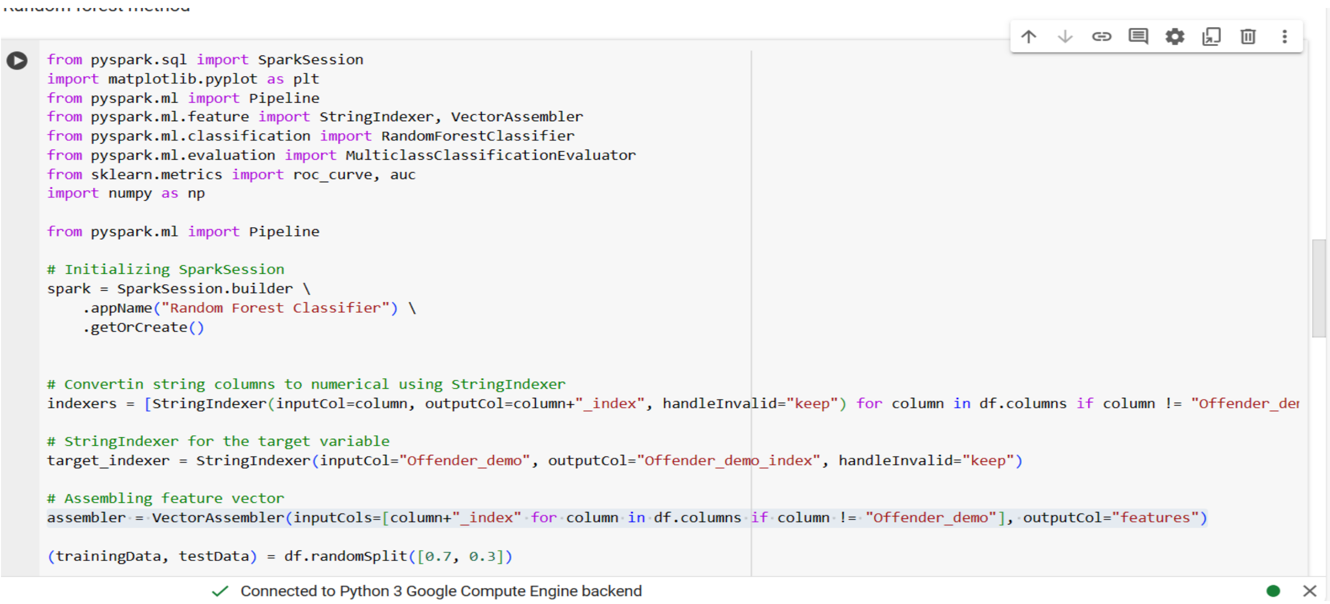
Next, we will import the dataset using read.csv function:

```
df = spark.read.csv('/content/cleaned_data.csv', header = True, inferSchema = True)
df.printSchema()
```

```
root
```

Data Modelling

Fistly , all the packages required to for the Random Forest Classifier modelling are loaded .The data set is split into training and test datasets. Two feature transformers StringIndexer and Vector Indexer are used to transform the data. These help in indexing the categories for the label and categorical features. StringIndexer encodes the categorical values and VectorIndexer combines multiple columns into single one. The meta data is added to the data frame.

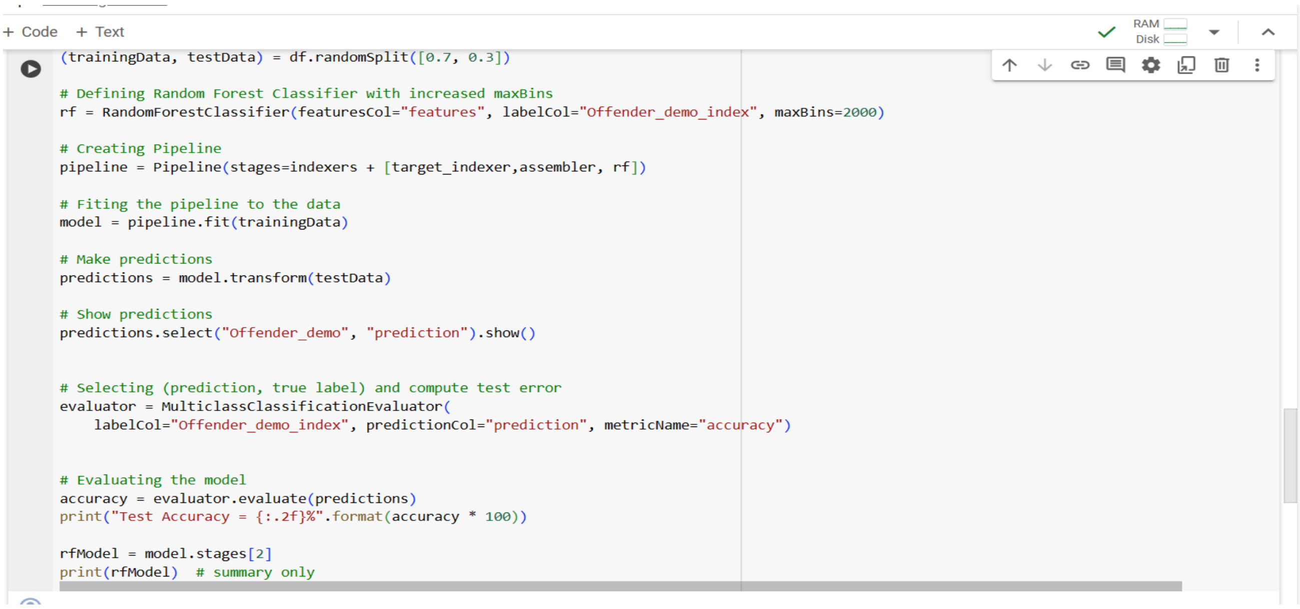


Random forest method

```
from pyspark.sql import SparkSession
import matplotlib.pyplot as plt
from pyspark.ml import Pipeline
from pyspark.ml.feature import StringIndexer, VectorAssembler
from pyspark.ml.classification import RandomForestClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from sklearn.metrics import roc_curve, auc
import numpy as np

from pyspark.ml import Pipeline

# Initializing SparkSession
spark = SparkSession.builder \
    .appName("Random Forest Classifier") \
    .getOrCreate()


# Convertin string columns to numerical using StringIndexer
indexers = [StringIndexer(inputCol=column, outputCol=column+"_index", handleInvalid="keep") for column in df.columns if column != "Offender_de

# StringIndexer for the target variable
target_indexer = StringIndexer(inputCol="Offender_demo", outputCol="Offender_demo_index", handleInvalid="keep")

# Assembling feature vector
assembler = VectorAssembler(inputCols=[column+"_index" for column in df.columns if column != "Offender_demo"], outputCol="features")

(trainingData, testData) = df.randomSplit([0.7, 0.3])
```

✓ Connected to Python 3 Google Compute Engine backend

Then the Random Forest classifier is defined with categorical values and the target variable.A pipeline is used for model tuning is created by adding the both feature transformers.The pipeline is fitted to the model which is trained with training data and the prediction is done on test data. Maxbin is set to 2000 since the data set had 1772 unique values in categories , the max bin was set beyond it for execution.The evaluator metric is to measure how well a fitted Model does on held out test data.

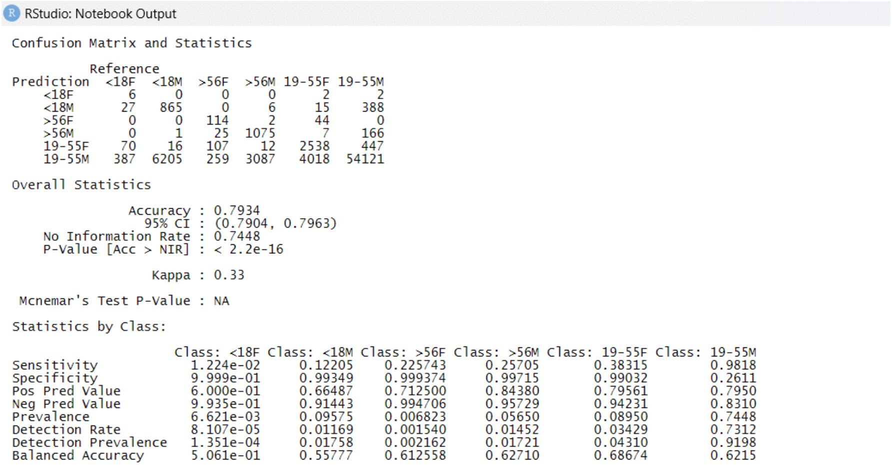


```
(trainingData, testData) = df.randomSplit([0.7, 0.3])

# Defining Random Forest Classifier with increased maxBins
rf = RandomForestClassifier(featuresCol="features", labelCol="Offender_demo_index", maxBins=2000)

# Creating Pipeline
pipeline = Pipeline(stages=indexers + [target_indexer,assembler, rf])

# Fiting the pipeline to the data
model = pipeline.fit(trainingData)

# Make predictions
predictions = model.transform(testData)

# Show predictions
predictions.select("Offender_demo", "prediction").show()


# Selecting (prediction, true label) and compute test error
evaluator = MulticlassClassificationEvaluator(
    labelCol="Offender_demo_index", predictionCol="prediction", metricName="accuracy")


# Evaluating the model
accuracy = evaluator.evaluate(predictions)
print("Test Accuracy = {:.2f}%".format(accuracy * 100))

rfModel = model.stages[2]
print(rfModel)  # summary only
```

6. Performance Evaluation And Comparison Of Methods

6.1 Perfomance Evaluation

The performance evaluation for Random Forest Classification in R :

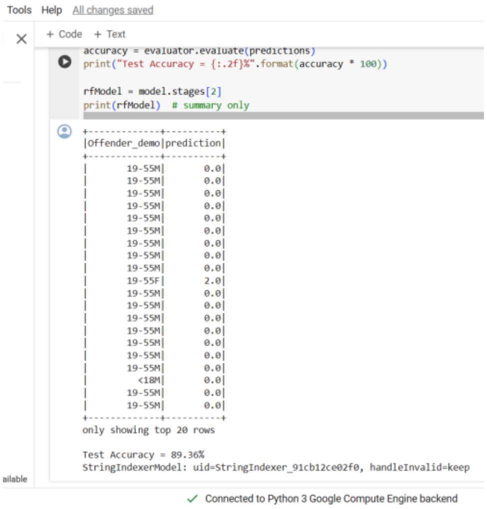


RStudio: Notebook Output

```
Confusion Matrix and Statistics

          Reference
Prediction  <18F  <18M  >56F  >56M 19-55F 19-55M
    <18F       6     0     0     0      2      2
    <18M      27   865     0     6     15    388
    >56F       0     0   114     2     44      0
    >56M       0     1    25  1075      7    166
    19-55F    70    16   107    12   2538    447
    19-55M   387  6205   259  3087   4018  54121

Overall Statistics

               Accuracy : 0.7934
                 95% CI : (0.7904, 0.7963)
    No Information Rate : 0.7448
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.33

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: <18F Class: <18M Class: >56F Class: >56M Class: 19-55F Class: 19-55M
Sensitivity            1.224e-02     0.12205    0.225743     0.25705        0.38315        0.9818
Specificity            9.999e-01     0.99349    0.999374     0.99715        0.99032        0.2611
Pos Pred Value         6.000e-01     0.66487    0.712500     0.84380        0.79561        0.7950
Neg Pred Value         9.935e-01     0.91443    0.994706     0.95729        0.94231        0.8310
Prevalence             6.621e-03     0.09575    0.006823     0.05650        0.08950        0.7448
Detection Rate         8.107e-05     0.01169    0.001540     0.01452        0.03429        0.7312
Detection Prevalence   1.351e-04     0.01758    0.002162     0.01721        0.04310        0.9198
Balanced Accuracy      5.061e-01     0.55777    0.612558     0.62710        0.68674        0.6215
```

An accuracy of 0.79 indicates that the Random Forest Model correctly predicted the target variable for approximately 79%.A kappa value 0.33 indicates that there is moderate agreement between the predicted classification and true classification. P value <0.05 suggest strong evidence to reject the null hypothesis. The confusion matrix displays the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) classifications for a multiclass classification problem. Each row corresponds to a predicted class and column corresponds to actual class. The cell at row **19-55M** and column **19-55M** contains the count of instances where the model predicted **19-55M** and the true label was also **19-55M**.

The performance evaluation of Random Forest in PySpark:



```
accuracy = evaluator.evaluate(predictions)
print("Test Accuracy = {:.2f}%".format(accuracy * 100))

rfModel = model.stages[2]
print(rfModel)  # summary only
```

```
+-------------+----------+
|Offender_demo|prediction|
+-------------+----------+
|       19-55M|       0.0|
|       19-55M|       0.0|
|       19-55M|       0.0|
|       19-55M|       0.0|
|       19-55M|       0.0|
|       19-55M|       0.0|
|       19-55M|       0.0|
|       19-55M|       0.0|
|       19-55M|       0.0|
|       19-55F|       2.0|
|       19-55M|       0.0|
|       19-55M|       0.0|
|       19-55M|       0.0|
|       19-55M|       0.0|
|       19-55M|       0.0|
|       19-55M|       0.0|
|       19-55M|       0.0|
|         <18M|       0.0|
|       19-55M|       0.0|
|       19-55M|       0.0|
+-------------+----------+
only showing top 20 rows

Test Accuracy = 89.36%
StringIndexerModel: uid=StringIndexer_91cb12ce02f0, handleInvalid=keep
```

The Random Forest model in PySpark gave a test accuracy of 89.36%. The Random Forest Model correctly predicted the target variable for approximately 89% and top 20 rows were displayed .

6.2 Comparison Of Methods

The test accuracy for the model were higher in Pyspark compared to R. R is single threaded and cannot scale well to large datasets whereas Pyspark effectively process large amounts of data by splitting up computations among several nodes.

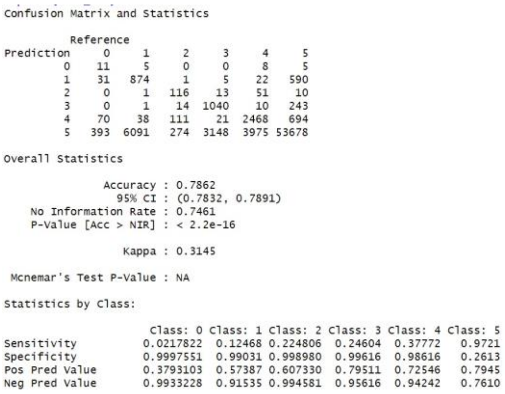
The memory capacity of R dealing with huge data set was low compared to that of PySpark ,certain issues were raised while allocation of 500 trees ,which led to the allocation of 300 trees to save the memory size.R performs well for smaller datasets, but has limitations when dealing with large data set such as the data set used in the project.

The execution time for training the dataset was long in R compared to that of in PySpark. It took around 50-60 minutes for random forest to train the model in R.For training the huge data set for just 100 trees R took long time enough. Rather PySpark executed it within 6 minutes.

7.Discussion of Findings

This section discusses the findings of other machine learning methods used on the data set Crime_Cleaned .

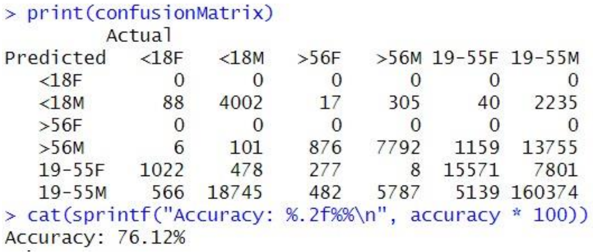
The method eXtreme Gradient Boostin produced the following results.



```
Confusion Matrix and Statistics

          Reference
Prediction     0     1     2     3     4     5
         0    11     5     0     0     8     5
         1    31   874     1     5    22   590
         2     0     1   116    13    51    10
         3     0     1    14  1040    10   243
         4    70    38   111    21  2468   694
         5   393  6091   274  3148  3975 53678

Overall Statistics

               Accuracy : 0.7862
                 95% CI : (0.7832, 0.7891)
    No Information Rate : 0.7461
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3145

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 0 Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity         0.0217822  0.12468 0.224806  0.24604  0.37772   0.9721
Specificity         0.9997551  0.99031 0.998980  0.99616  0.98616   0.2613
Pos Pred Value      0.3793103  0.57387 0.607330  0.79511  0.72546   0.7945
Neg Pred Value      0.9933228  0.91535 0.994581  0.95616  0.94242   0.7610
```

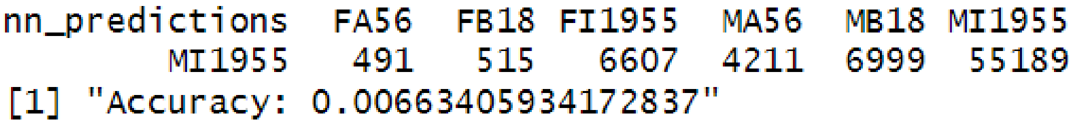
The accuracy of eXtreme Gradient Boosting was around 0.78 and kappa =0.31 indicating a moderate value and p<0.05 rejecting a null hypothesis.The

The result of Naïve Bayes Method produced an accuracy values of 0.76 which is moderate and the confusion matrix presents the results of predicted values versus the actual values. Most predictions for categories like <18M and 19-55M are high suggesting these predictions were correctly identified and predicted. The pyspark test accuracy value is 0.19



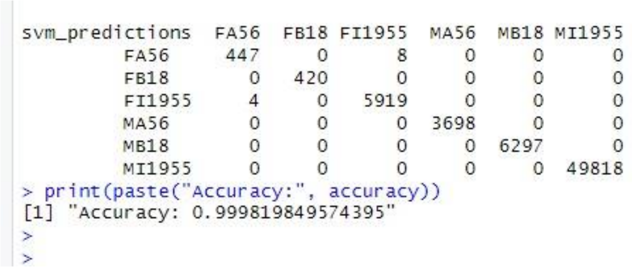
```
> print(confusionMatrix)
          Actual
Predicted   <18F   <18M   >56F   >56M 19-55F 19-55M
    <18F       0      0      0      0      0      0
    <18M      88   4002     17    305     40   2235
    >56F       0      0      0      0      0      0
    >56M       6    101    876   7792   1159  13755
    19-55F  1022    478    277      8  15571   7801
    19-55M   566  18745    482   5787   5139 160374
> cat(sprintf("Accuracy: %.2f%%\n", accuracy * 100))
Accuracy: 76.12%
```

The results of Neural Network produced an accuracy of 0.0066



```
nn_predictions   FA56   FB18 FI1955   MA56   MB18 MI1955
        MI1955    491    515   6607   4211   6999  55189
[1] "Accuracy: 0.00663405934172837"
```

The result of Support Vector Machine produced the following results :



```
svm_predictions  FA56  FB18 FI1955  MA56  MB18 MI1955
         FA56     447     0      8     0     0      0
         FB18       0   420      0     0     0      0
         FI1955     4     0   5919     0     0      0
         MA56       0     0      0  3698     0      0
         MB18       0     0      0     0  6297      0
         MI1955     0     0      0     0     0  49818
> print(paste("Accuracy:", accuracy))
[1] "Accuracy: 0.999819849574395"
>
>
```

SVM method produced an accuracy of 0.99 .

From the discussion of above findings ,it can be concluded that all the above machine learning methods were used to model the Crime_cleaned data set and predicted an accuracy in the range of 0.75-0.99 and kappa values suggesting moderate agreement and p value<0.05 indicating a strong evidence in rejecting the null hypothesis and most predictions were predicted correctly for 19-55M .Hence it can be analysed from the predictions that offender 's demographic that is the age and gender fall in the category of 19-55M in the context of homicides in US. The accuracy scores of Random Forest ,eXtreme Gradient Boosting had similar scores in the range of 0.76 to 0.79.The method Support Vector Machine produced an higher accuracy for the model.The method Neural Networks gave a less accuracy score for the model.

The objective of the report to generate insights and value by processing of a dataset by applying various analytical methoda have been successfully.The data set was collected , prepared and cleaned.Exploratory data analysis was performed on the data set also using an unsupervised learning method .Random Forest classification method was used to model the data and a high computationa method Pyspark was used to run the model .The performance of both methods has been evaluated and discussed .The findings of various other methods of machine learning on the data set has also been discussed with respect to the research question of finding offender's demographic .

8. DMP AND AUTHORSHIP

T.M.M.Jayaweera, Kishan Kumar SenthilKumaran, Palash Joshi, Keerthana K.P, Tejas Kadam developed the research topic ,collected data , did the data preparation and performed exploratory analysis. T.M.M.Jayaweera introduced eXtreme Gradient Boosting, Keerthan KP implemented Random Forest Classification, Palash Joshi implemented Naïve Bayes , Tejas Kadam implemented Support Vector Machine and Kishan Kumar SenthilKumaran implemented Neural Network.